

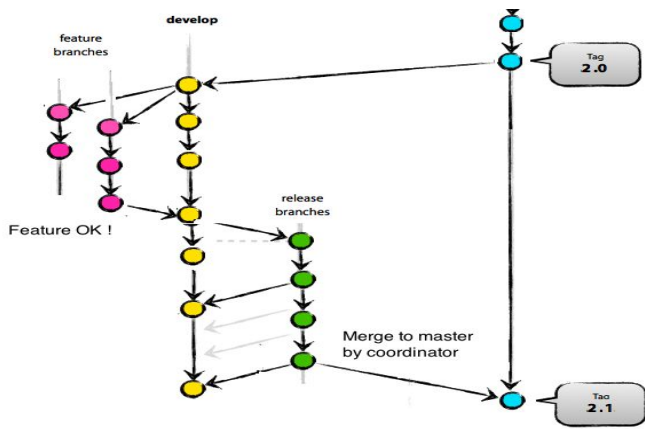


INFORMATION TECHNOLOGY INNOVATE

Collaboration Guide

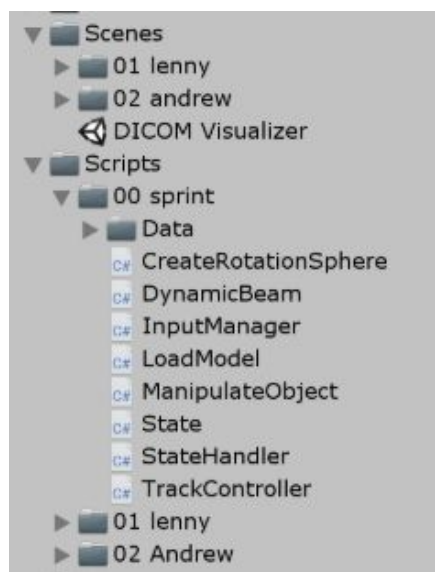
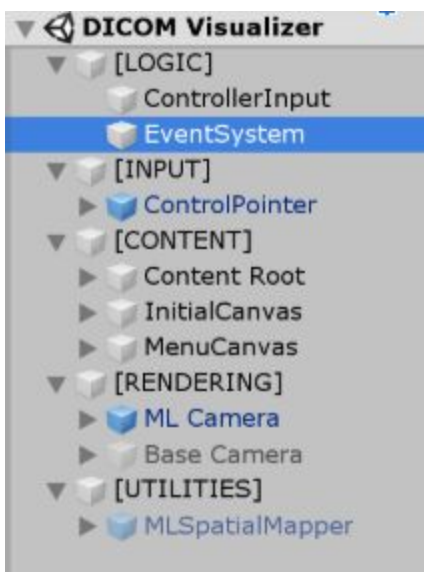
Version Control

- Set-up separate GitHub repo from UMIinnovate
- Every sprint starts with a new branch off of `master`.
 - Call it `dev#` or `sprint#`.
 - This is your home (dev) branch for the rest of the sprint.
- Every JIRA task/bug should involve adding a feature/fix branch off the `dev` branch.
 - Name it appropriately (i.e. "load_from_url" or by JIRA issue like "DV-40" for dicom visualizer issue 40).
 - Once the feature/fix is done make a pull request to the 'dev' branch and have a teammate review via GitHub Interface.
 - Will slow things down, but keep team in the loop. Reviewer(s) should highlight and comment where appropriate. If things are good, Reviewers should approve before they merge the request.
- Every sprint ends with a merge between `dev` and `master`.
 - Pull request is made and should be reviewed.
 - Rinse & Repeat



Unity & Scripting Golden Rules

1. Instantiate different scenes
 - a. each person has their own folder (i.e. "02 Andrew")
2. When editing prefabs create variants
 - a. each member has their own prefabs to edit
3. Consistent coding standard (makes reading code easier)
4. Create alternative scripts when experimenting
5. Consistent scene hierarchy standard (ease-of-use)



Teamwork Standards

1. Update (frequently)
 - a. Keep rest of the team informed on where-abouts and activity
 - b. Even saying "hello" can go a long way
2. Document (periodically)
 - a. Agile involves speedy dev, so keep it simple (detailed if possible)
 - b. Photos, audio, video, text
 - c. Future-proofing for yourself and others
 - d. "Speaking only helps who's in the room, writing helps everyone. This includes people who couldn't make it, or future employees who join years from now"
3. Review (code, sketches, concepts)
 - a. Helps each member stay in the loop
 - b. Chance one member finds an issue that was left unspotted
 - c. Promotes discussion among team
 - d. GitHub Interface makes commenting on code snippets easy

The screenshot shows a GitHub pull request interface. At the top, a comment from AndrewXR states: "Lenny and Andrew removed obsolete WWW Unity class and replaced with UnityWebRequest. Additionally, running web request via a coroutine for efficiency." Below this, a commit history shows four commits by lennymartinez: "add a dictionary for spelt words", "update readme", "Update Loading to use UnityWebRequest.", and "Refactored WebRequest to use CoRoutine and UnityWebRequest". A green checkmark indicates that lennymartinez approved these changes. The code diff for the file "dicom-visualizer/Assets/Plugins/Lumin/MagicLeapManifestSettings.asset" is shown, with a change in the "m_MinimumAPILevel" property from 7 to 6. A comment from lennymartinez asks: "Shouldn't this be 7? I think you might want to check your settings".

AndrewXR commented 4 days ago

Lenny and Andrew removed obsolete WWW Unity class and replaced with UnityWebRequest. Additionally, running web request via a coroutine for efficiency.

lennymartinez and others added 4 commits 8 days ago

- add a dictionary for spelt words
- update readme
- Update Loading to use UnityWebRequest.
- Refactored WebRequest to use CoRoutine and UnityWebRequest

lennymartinez approved these changes 4 days ago

dicom-visualizer/Assets/Plugins/Lumin/MagicLeapManifestSettings.asset

```
@@ -12,7 +12,7 @@ MonoBehaviour:
12 12     m_Script: {fileID: 11500000, guid: 9cf5b356e131a944286fec26d7c40b4e, type:
13 13     m_Name: MagicLeapManifestSettings
14 14     m_EditorClassIdentifier:
15 -     m_MinimumAPILevel: 7
15 +     m_MinimumAPILevel: 6
```

lennymartinez 4 days ago

Shouldn't this be 7? I think you might want to check your settings